

**IMPLEMENTATION OF A UNIFIED DEVELOPER COLLABORATION
PLATFORM FOR INTEGRATING CODE MANAGEMENT AND API TESTING
TOOLS USING OPEN SOURCE TECHNOLOGIES**

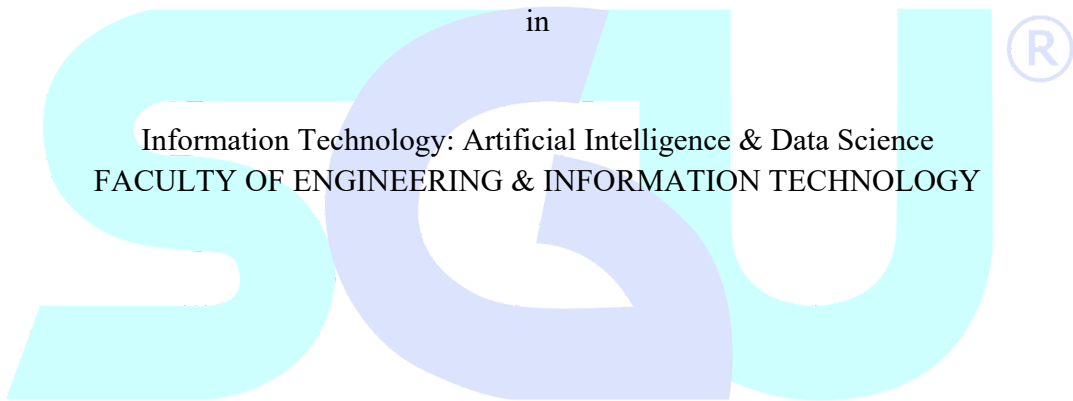
By

Lidya Aurelcia
12202034

BACHELOR'S DEGREE

in

Information Technology: Artificial Intelligence & Data Science
FACULTY OF ENGINEERING & INFORMATION TECHNOLOGY



SWISS GERMAN UNIVERSITY
The Prominence Tower
Jalan Jalur Sutera Barat No. 15, Alam Sutera
Tangerang, Banten 15143 - Indonesia

June 2026

STATEMENT BY THE AUTHOR

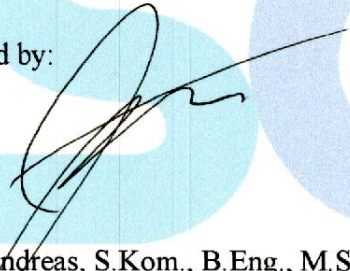
I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgement is made in the thesis.


Lidya Aurelcia

Student

22 June 2026
Date

Approved by:


Joseph Andreas, S.Kom., B.Eng., M.Sc.

Thesis Advisor

24 JUNE 2026
Date


Leonardo Theoson, S.Kom., B.Eng.

Thesis Co-Advisor

22 June 26
Date

Dr. Maulahikmah Galinium, S.Kom., M.Sc.

Dean

Date

Lidya Aurelcia

ABSTRACT

IMPLEMENTATION OF A UNIFIED DEVELOPER COLLABORATION PLATFORM FOR INTEGRATING CODE MANAGEMENT AND API TESTING TOOLS USING OPEN SOURCE TECHNOLOGIES

By

Lidya Aurelcia

Joseph Andreas, S.Kom., B.Eng., M.Sc.*

Leonardo Theoson, S.Kom., B.Eng.**

SWISS GERMAN UNIVERSITY*

CODEDOC INDONESIA**

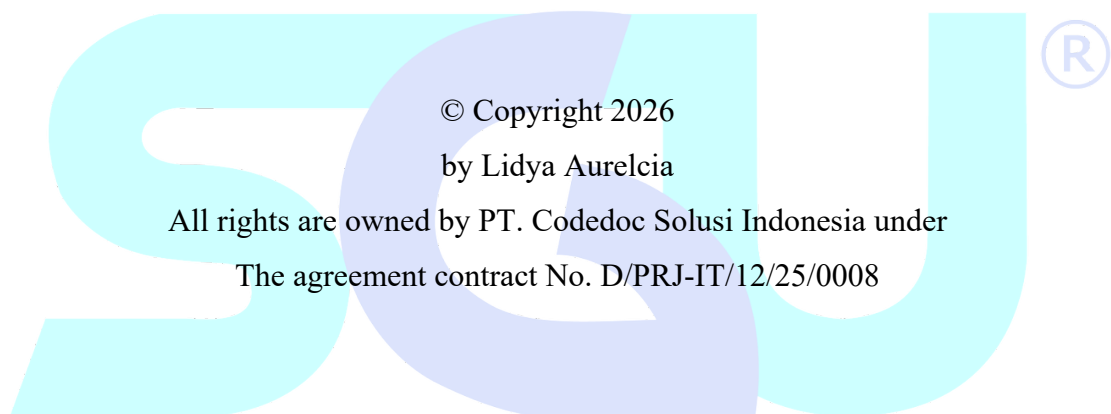
This research aimed to develop and evaluate a middleware-based integration platform that connected repository management and API collection management workflows using GitLab CE and Hoppscotch. The platform was designed to reduce manual coordination when developers needed to keep repository branches and API collection artifacts aligned. The research used a controlled company testing environment and applied comparative workflow evaluation, scenario-based User Acceptance Testing, a post-UAT questionnaire, and orchestration latency measurement. The evaluation was also aligned with selected COBIT 2019 objectives to assess configuration alignment, change traceability, workflow quality, review support, and performance monitoring.

The findings showed that the integrated workflow reduced the average manual action count from 14.6 actions to 10 actions and decreased the average tool-switching count from 2.2 switches to 2 switches. The UAT results showed that the repository–collection synchronization workflow passed in both personal and group project contexts. The questionnaire results indicated positive user acceptance, while latency testing showed that all sampled middleware

operations met the defined response-time threshold. Therefore, the proposed platform was feasible for supporting repository–collection synchronization in the tested environment.

Keywords: middleware integration, repository management, API collection, workflow synchronization, COBIT 2019.





© Copyright 2026

by Lidya Aurelcia

All rights are owned by PT. Codedoc Solusi Indonesia under

The agreement contract No. D/PRJ-IT/12/25/0008

DEDICATION

This thesis is dedicated to my family, whose support, patience, and encouragement have helped me throughout my academic journey. I also dedicate this work to the people who believed in me during difficult moments and reminded me to keep going. Their presence and support have been an important part of completing this study.

ACKNOWLEDGEMENTS

First and foremost, my heartfelt thanks go to my family for their endless support, prayers, understanding, and encouragement during the process of completing this thesis. Their support gave me the motivation and grit to continue even during the most challenging parts.

I would like to express my deepest gratitude to both my thesis advisor, Joseph Andreas, S.Kom., B.Eng., M.Sc., and my co-advisor, Leonardo Theoson, S.Kom., B.Eng., for their continuous guidance, patience, and support throughout the completion of this research. Their willingness to go above and beyond in providing feedback, direction, and encouragement has truly been invaluable.

I am sincerely grateful to PT Codedoc Solusi Indonesia for allowing this research to be conducted in the first place. I am also thankful for to all colleagues in the company, and also the three much-beloved kitties, Soya, Tofu, and Pepe for providing an environment that I feel accepted, secure, and can thrive in. I would also like to thank all my friends and all (including former) students of BIT 2022, for their encouragement and support throughout this journey. Their presence made the process all the more meaningful.

Finally, I acknowledge that this thesis would not have been completed without the help, guidance, and support of many people. Although I may not be able to mention everyone individually, I am truly grateful for every contribution, advice, and encouragement that helped me reach this point.

TABLE OF CONTENTS

STATEMENT BY THE AUTHOR	2
ABSTRACT	3
DEDICATION	6
ACKNOWLEDGEMENTS	7
TABLE OF CONTENTS	8
LIST OF FIGURES	11
LIST OF TABLES	12
CHAPTER 1 – INTRODUCTION.....	13
1.1 Background.....	13
1.2 Problem Statement.....	14
1.3 Research Objectives	14
1.4 Research Scope.....	15
1.5 Research Limitation.....	15
1.6 Research Questions	15
1.7 Hypothesis	16
1.8 Significance of the Study.....	16
CHAPTER 2 – LITERATURE REVIEW.....	17
2.1 Repository Management Tools.....	17
2.1.1 Gitea	17
2.1.2 GitLab Community Edition	17
2.1.3 GitHub	18
2.1.4 Bitbucket.....	18
2.1.5 Comparison Between Repository Management Tools	18
2.2 Application Programming Interface (API) Testing Tools.....	19
2.2.1 Hoppscotch	19
2.2.2 Insomnia	20
2.2.3 SoapUI.....	20
2.2.4 Postman	20
2.2.5 Comparison Between API Testing Tools	21

2.3	Web Application Development Technologies.....	21
2.3.1	Next.js.....	22
2.3.2	Bun	22
2.4	Secure Development Life Cycle (SDLC) Models.....	22
2.5	Control Objectives for Information and Related Technologies (COBIT) 2019.....	25
2.6	Related Works	28
2.6.1	Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective (Waseem et al., 2021)	28
2.6.2	Systematic Mapping Study of Test Generation for Microservices: Approaches, Challenges, and Impact on System Quality (Miao, Shaafi and Song, 2025)	28
2.6.3	Designing Developer Platforms for Cross-Cloud Portability and Scale (Kothandarama, 2025).....	29
2.6.4	Monitoring Tools for DevOps and Microservices: A Systematic Grey Literature Review (Giamattei et al., 2024).....	30
2.6.5	Microservice API Evolution in Practice: A Study on Strategies and Challenges (Lercher et al., 2024)	30
CHAPTER 3 – RESEARCH METHODOLOGY		33
3.1	Research Overview.....	33
3.2	Requirement Analysis	33
3.2.1	Functional Requirements.....	33
3.2.2	Non-Functional Requirements.....	34
3.2.3	System Requirements	35
3.3	System Design	36
3.3.1	Architecture Diagram	36
3.3.2	Use Case Diagram	39
3.3.3	Activity Diagram	41
3.4	Development and Verification	44
3.5	Evaluation Phase Testing	44
3.5.1	Comparative End-to-End Workflow Evaluation	45
3.5.2	Scenario-Based User Acceptance Testing.....	49
3.5.3	Orchestration Latency Evaluation	57
3.6	Time Frame	59
CHAPTER 4 – RESULTS AND DISCUSSION		60
4.1	Comparative End-To-End Workflow Measurement Results	60
4.2	Scenario-Based User Acceptance Testing Results	62
4.3	Orchestration Latency Evaluation Results	70

4.4	Results Summary.....	71
CHAPTER 5 – CONCLUSIONS AND RECOMMENDATIONS.....		73
5.1	Conclusion.....	73
5.2	Recommendations For Future Works.....	74
GLOSSARY		75
REFERENCES		76
APPENDIX		78



LIST OF FIGURES

Figure 3.1 Research Overview Diagram	33
Figure 3.2 Overall Architecture Diagram.....	36
Figure 3.3 Local Architecture Diagram.....	37
Figure 3.4 Use Case Diagram.....	39
Figure 3.5 Activity Diagram.....	41
Figure 3.6 Timeline of This Study	59



LIST OF TABLES

Table 2.1 Comparison Between Repository Management Tools (Atlassian, 2025; Gitea, no date.; GitHub, 2025; Gitlab, 2025).....	18
Table 2.2 Comparison Between API Testing Tools (Hoppscotch, no date.; Kong, 2025; Postman, 2025; SmartBear, 2025).....	21
Table 2.3 Comparison Between SDLC Models (Diansyah et al., 2023).....	23
Table 2.4 COBIT 2019 and its relation to this study.....	25
Table 2.5 Comparison Between Works.....	31
Table 3.1 Minimum and Recommended System Requirements	35
Table 3.2 Comparative End-to-End Workflow Evaluation Metric and Measurement Method	46
Table 3.3 Counting Rules and Metrics for Comparative End-to-End Workflow Evaluation .	47
Table 3.4 UAT Scenarios	50
Table 3.5 Post-UAT Questionnaire	55
Table 3.6 Latency Test Operations Tested and Logged Measurement	57
Table 4.1 Workflow Measurement Results for Baseline and Integrated Approaches.....	60
Table 4.2 Statistical Summary of Manual Action Count	61
Table 4.3 Statistical Summary of Tool-Switching Count	61
Table 4.4 Aggregated Scenario-Based UAT Results	63
Table 4.5 Aggregated Post-UAT Questionnaire Result	66
Table 4.6 Post-UAT Questionnaire Results by COBIT Category.....	69
Table 4.7 Aggregated Orchestration Latency Results	70
Table 4.8 Summary of Evaluation Evidence, Research Questions Answered, and Related COBIT Metrics	71